

# ΣΧΕΔΙΑΣΜΌΣ ΣΥΣΤΗΜΆΤΩΝ VLSI

## Τρίτη Εργαστηριακή Άσκηση

Γιώργος Ντάκος 1059569

27 Απριλίου 2021

### Περίληψη

Στο παρακάτω κείμενο παραδίδεται η αναφορά της 3<sup>ης</sup> εργαστηριακής άσκησης του μαθήματος Σχεδιασμός Συστημάτων VLSI. Όλοι οι κώδικες έχουν συγγραφεί σε γλώσσα **VHDL** μέσω του **Notepad++** και έχουν μεταγλωττιστεί και επιβεβαιωθεί οι λειτουργίες τους μέσω του εργαλείου **ModelSim**. Τέλος η σύνθεση όλων των κυκλωμάτων πραγματοποιήθηκαν από το εργαλείο **VIVADO** της **Xilinx**.

## Περιεχόμενα

Ενότητα Α	4
Ενότητα Β	5
Κώδικες σε VHDL	10
Βιβλιογραφία	13

## Κατάλογος Σχημάτων

1	Μνήμη RAM . . . . .	6
2	Simulation της RAM . . . . .	7
3	Simulation της RAM36B . . . . .	7

## Κατάλογος Πινάκων

1	Αποτελέσματα Σύνθεσης . . . . .	4
2	Αποτελέσματα Σύνθεσης RAM Σχεδιαμών . . . . .	8

## Ενότητα Α

### Α.1,Α.2

<i>Circuit</i>	<i>LUTs</i>	<i>IO</i>	<i>FlipFlops</i>	<i>BRAM</i>	<i>Critical Path (ns)</i>	<i>Clock (MHz)</i>
nand64	64(0.31%)	192(91.43%)	0	0	5.231	0
nand64me	64(0.31%)	192(91.43%)	0	0	4.231	0
full_adder	1(0.00%)	5(02.38%)	0	0	4.337	0
logic_unit	8(0.04%)	32(15.24%)	0	0	4.337	0
logic_unit_reg	9(0.04%)	33(15.71%)	16	0	1.507	100
mux_4_to_1	9(0.04%)	26(12.38%)	0	0	4.754	0
second_mux_4_to_1	9(0.04%)	26(12.38%)	0	0	4.754	0
circuitA	1(0.00%)	4(01.90%)	0	0	4.321	0
circuitB	1(0.00%)	4(01.90%)	0	0	4.321	0

Table 1: Αποτελέσματα Σύνθεσης

Η σύνθεση των κυκλωμάτων των Εργαστηριακών Ασκήσεων 1 και 2 πραγματοποιήθηκε σε **FPGA** οικογένειας **Artix-7**. Τα περισσότερα από τα εξεταζόμενα κυκλώματα είναι καθαρά συνδυαστικά, ενώ μόνο το κύκλωμα **logic\_unit\_reg** περιλαμβάνει διαδοχική λογική και χρονισμό με ρολόι. Για τον λόγο αυτό, τα κυκλώματα αξιολογούνται με διαφορετική χρονική μετρική, ανάλογα με τη φύση τους. Για τα συνδυαστικά κυκλώματα, η απόδοση εκτιμήθηκε βάση του μέγιστου χρόνου διάδοσης (*critical path delay*) από είσοδο σε έξοδο, όπως αυτός προκύπτει από το *Datasheet Combinational Timing* (pin-to-pin, slow process corner). Για το χρονισμένο κύκλωμα, η απόδοση αξιολογήθηκε βάσει του χειρότερου μονοπατιού εισόδου προς καταχωρητή (*input-to-register*), με χρονικό περιορισμό **100 MHz**.

Τα κυκλώματα **nand64** και **nand64me** παρουσιάζουν ίδιο αριθμό λογικών πόρων (64 LUTs) και ίδιο αριθμό I/O (192), γεγονός απολύτως αναμενόμενο, καθώς υλοποιούν την ίδια λειτουργία: 64 ανεξάρτητες πύλες NAND σε bitwise μορφή. Ο υψηλός αριθμός I/O οφείλεται αποκλειστικά στο μεγάλο πλάτος των διαύλων εισόδου και εξόδου και όχι στην πολυπλοκότητα της λογικής. Η μικρή διαφορά στον αναφερόμενο χρόνο διάδοσης μεταξύ των δύο υλοποιήσεων αποδίδεται σε διαφορετικές αποφάσεις βελτιστοποίησης και δρομολόγησης του εργαλείου σύνθεσης και δεν επηρεάζει το γενικό συμπέρασμα ότι τα δύο κυκλώματα είναι ισοδύναμα από πλευράς επιφάνειας και λειτουργικότητας.

Ο **full\_adder** αποτελεί ένα πολύ μικρό συνδυαστικό κύκλωμα και υλοποιείται με ελάχιστους πόρους (1 LUT), γεγονός που καταδεικνύει την ικανότητα του εργαλείου σύνθεσης να συμπτύσσει απλή λογική σε έναν μόνο πίνακα αλήθειας. Ο αριθμός I/O (5) είναι περιορισμένος και αντιστοιχεί άμεσα στη διεπαφή του κυκλώματος. Ο αναφερόμενος χρόνος διάδοσης είναι pin-to-pin και περιλαμβάνει τις καθυστερήσεις των I/O buffers, χωρίς να υποδηλώνει μεγάλο βάθος λογικής.

Το κύκλωμα **logic\_unit** υλοποιεί λογικές πράξεις AND και XOR σε δεδομένα πλάτους 8 bit και για τον λόγο αυτό απαιτεί περισσότερους LUTs (8) σε σχέση με τον full adder. Ο αριθμός I/O (32) είναι αναμενόμενος, καθώς περιλαμβάνει δύο 8-bit εισόδους και δύο 8-bit εξόδους. Ο χρόνος διάδοσης παραμένει συγκρίσιμος με άλλα μικρά συνδυαστικά κυκλώματα, δεδομένου ότι αξιολογείται με βάση πλήρη pin-to-pin διαδρομή.

Το `logic_unit_reg` αποτελεί τη χρονισμένη εκδοχή του προηγούμενου κυκλώματος και είναι το μοναδικό κύκλωμα που περιλαμβάνει ακολουθιακή λογική. Η παρουσία 16 flip-flops αντιστοιχεί άμεσα στις δύο 8-bit καταχωρημένες εξόδους. Ο αριθμός LUTs αυξάνεται οριακά (9), γεγονός αναμενόμενο λόγω της προσθήκης καταχωρητών και της σχετικής συνδεσμολογίας. Ο κρίσιμος χρόνος διάδοσης (1.507 ns) αφορά τη χειρότερη διαδρομή εισόδου προς καταχωρητή και είναι σημαντικά μικρότερος από τους pin-to-pin χρόνους διάδοσης των συνδυαστικών κυκλωμάτων. Με χρονισμό 100 MHz (περίοδος 10 ns), το κύκλωμα λειτουργεί με άνετο χρονικό περιθώριο.

Τα κυκλώματα `mux_4_to_1` και `second_mux_4_to_1` παρουσιάζουν πανομοιότυπα αποτελέσματα σε πόρους και καθυστέρηση, καθώς υλοποιούν την ίδια λειτουργία πολυπλεξίας με διαφορετικό συντακτικό VHDL. Το εργαλείο σύνθεσης αναγνωρίζει τη λογική ισοδυναμία των περιγραφών και παράγει το ίδιο τελικό hardware.

Τέλος, τα `circuitA` και `circuitB`, παρότι περιγράφονται με διαφορετικό στυλ (behavioral και δομικό αντίστοιχα), οδηγούν σε πανομοιότυπα αποτελέσματα σύνθεσης. Ο ίδιος αριθμός LUTs, I/O και χρόνος διάδοσης καταδεικνύουν ότι το εργαλείο σύνθεσης βασίζεται στη λογική συνάρτηση και όχι στο συντακτικό της περιγραφής, καταλήγοντας σε ισοδύναμο κύκλωμα.

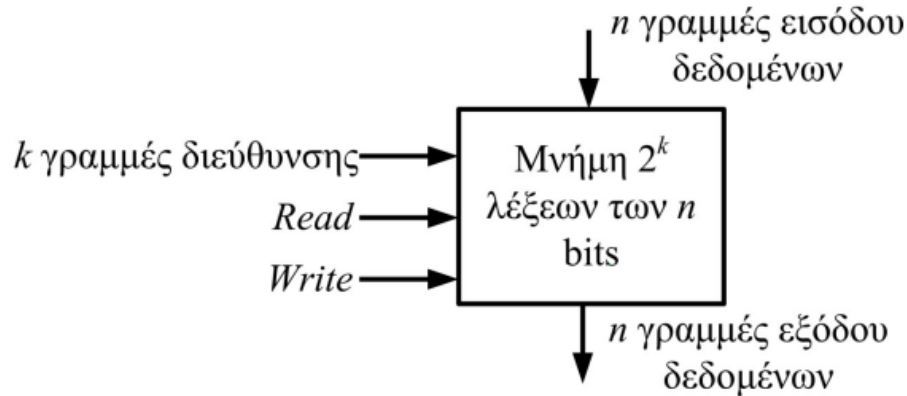
Συνολικά, τα αποτελέσματα σύνθεσης είναι συνεπή με τη θεωρία και τις προσδοκίες. Η κατανάλωση πόρων εξαρτάται κυρίως από το πλάτος των δεδομένων και τη λειτουργική πολυπλοκότητα, ενώ η χρονική απόδοση διαφοροποιείται ανάλογα με το αν το κύκλωμα είναι συνδυαστικό ή χρονισμένο. Επιπλέον, επιβεβαιώνεται ότι διαφορετικές περιγραφές της ίδιας λογικής οδηγούν σε ισοδύναμο hardware, ενώ η εισαγωγή καταχωρητών βελτιώνει τη χρονική συμπεριφορά με μικρό κόστος σε επιφάνεια.

## Ενότητα Β

### B.1,B.2

Η λειτουργία της μνήμης 32×8 που υλοποιήθηκε σε καθαρή VHDL επαληθεύτηκε μέσω προσομοίωσης. Στο testbench εφαρμόστηκε αρχικά reset, το οποίο αρχικοποιεί τις πρώτες θέσεις μνήμης με γνωστές τιμές. Στη συνέχεια εφαρμόστηκε σήμα εγγραφής (WR=1) και γράφτηκε η τιμή 0x05 στη διεύθυνση 0x04. Στο επόμενο ανερχόμενο μέτωπο του ρολογιού, η τιμή αυτή αποθηκεύτηκε στη μνήμη. Ακολούθως το WR τέθηκε στο 0 (λειτουργία ανάγνωσης) και στη διεύθυνση 0x04 παρατηρήθηκε στην έξοδο DOUT η τιμή 0x05, επιβεβαιώνοντας την ορθή λειτουργία της εγγραφής και ανάγνωσης.

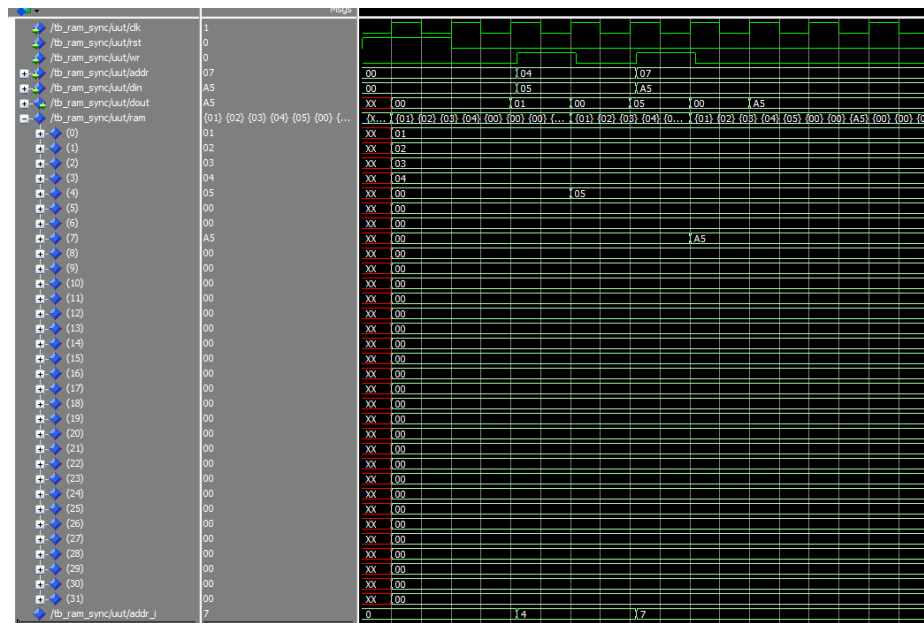
Η ίδια διαδικασία επαναλήφθηκε και για δεύτερη διεύθυνση (0x07) με τιμή 0xA5, η οποία επίσης ανακτήθηκε σωστά από τη μνήμη. Τα αποτελέσματα αυτά επιβεβαιώνουν ότι η VHDL RAM λειτουργεί ως σύγχρονη μνήμη μονής θύρας με μία περίοδο καθυστέρησης ανάγνωσης.



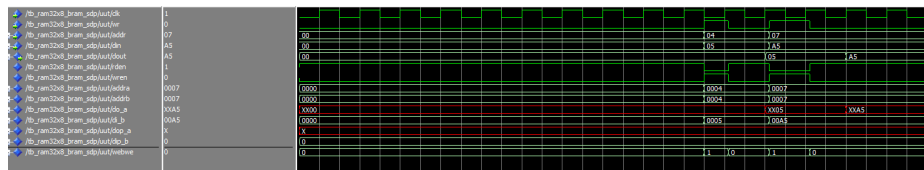
Σχήμα 1: Μνήμη RAM

Η υλοποίηση της μνήμης με χρήση του ενσωματωμένου Block RAM (RAMB18E1) επαληθεύτηκε μέσω προσομοίωσης με το ίδιο testbench. Αρχικά εφαρμόστηκε reset ώστε να καθοριστεί γνωστή αρχική κατάσταση. Στη συνέχεια, με ενεργοποιημένο το σήμα εγγραφής ( $WR=1$ ), η τιμή 0x05 γράφτηκε στη διεύθυνση 0x04 μέσω της θύρας εγγραφής της BRAM. Στο επόμενο clock edge, η τιμή αποθηκεύτηκε στο Block RAM.

Όταν το  $WR$  τέθηκε στο 0 (λειτουργία ανάγνωσης), η ίδια διεύθυνση 0x04 εφαρμόστηκε στη θύρα ανάγνωσης και στην έξοδο DOUT εμφανίστηκε η τιμή 0x05 μετά από ένα clock cycle, όπως προβλέπεται για σύγχρονη μνήμη. Η διαδικασία επαναλήφθηκε για δεύτερη διεύθυνση (0x07) με δεδομένο 0xA5, το οποίο ανακτήθηκε σωστά. Τα αποτελέσματα αυτά επιβεβαιώνουν ότι η υλοποίηση με Block RAM λειτουργεί σωστά και παρουσιάζει τη χαρακτηριστική χρονική καθυστέρηση μίας περιόδου κατά την ανάγνωση.



Σχήμα 2: Simulation της RAM



Σχήμα 3: Simulation της RAM36B

### B.3,B.4

Στο παρόν ερώτημα συγκρίθηκαν δύο υλοποιήσεις της ίδιας μνήμης  $32 \times 8$  bytes: (α) μία καθαρά συνθετική υλοποίηση σε VHDL με χρήση καταχωρητών και λογικής LUT, και (β) μία υλοποίηση με χρήση ενσωματωμένης μνήμης Block RAM (RAMB18E1) του FPGA Artix-7.

Τα αποτελέσματα της σύνθεσης δείχνουν ότι η υλοποίηση με Block RAM χρησιμοποιεί σχεδόν μηδενικούς πόρους γενικής λογικής (LUTs Flip-Flops), ενώ δεσμεύει μόνο 0.5 Block RAM, σε αντίθεση με την VHDL υλοποίηση που καταναλώνει 104 LUTs και 264 Flip-Flops. Το αποτέλεσμα αυτό επιβεβαιώνει τον αρχικό σχεδιασμό, σύμφωνα με τον οποίο οι ενσωματωμένες μνήμες του FPGA είναι ειδικά σχεδιασμένες για αποθήκευση δεδομένων και είναι πολύ πιο αποδοτικές σε πόρους από τη λογική υλοποίηση μνήμης με καταχωρητές.

Όσον αφορά την απόδοση, η LUT-based RAM παρουσιάζει κρίσιμο μονοπάτι 2.15 ns και μέγιστη συχνότητα λειτουργίας περίπου 465 MHz, ενώ η υλοποίηση με Block RAM εμφανίζει μικρότερο κρίσιμο μονοπάτι περίπου 1.85 ns και μέγιστη συχνότητα περίπου 540 MHz. Η βελτίωση αυτή είναι αναμενόμενη, καθώς στην υλοποίηση με Block RAM το κρίσιμο μονοπάτι περιλαμβάνει κυρίως το εσωτερικό clock-to-output delay του RAMB18E1 και όχι γενική λογική LUT.

Σύμφωνα με το επίσημο datasheet του Artix-7 (DS181), ο χρονικός παράγοντας clock-to-output της μνήμης RAMB18E1 χωρίς καταχωρημένη έξοδο (TRCKO\_DO) για speed grade 3 είναι 1.85 ns (μέγιστη τιμή) [1]. Η τιμή αυτή αντιστοιχεί σε θεωρητική μέγιστη συχνότητα περίπου 540 MHz και συμφωνεί με τις τιμές που παρατηρήθηκαν στη σύνθεση του ίαδο.

Επιπλέον, το Xilinx Memory Resources User Guide (UG473) επιβεβαιώνει ότι οι θύρες της Block RAM είναι πλήρως σύγχρονες και οι έξοδοι δεδομένων παράγονται μετά από συγκεκριμένο χρονικό διάστημα clock-to-output, το οποίο καθορίζεται από τις προδιαγραφές του datasheet [2]. Συνεπώς, το γεγονός ότι η BRAM υλοποίηση εμφανίζει καλύτερη απόδοση και μικρότερη κατανάλωση πόρων είναι πλήρως αναμενόμενο και επιβεβαιώνει την ορθότητα του αρχικού σχεδιασμού.

Circuit	LUTs	IO	FlipFlops	BRAM	Critical Path (ns)	Clock (MHz)	Fmax(MHz)
RAM 32x8	104(0.5%)	24(11.43%)	264(0.63%)	0.0(0%)	2.15	100.00	465.00
RAM32x8.BRAM.SDP	1(0.00%)	23(10.95%)	0(0.00%)	0.5(1%)	1.85	100.00	541.60

Table 2: Αποτελέσματα Σύνθεσης RAM Σχεδιαμών



Όπως βλέπουμε η πρώτη μνήμη μας αξιοποίησε από το FPGA αρκετά ικανοποιητικούς πόρους και θα λέγαμε ότι είναι κάτι που αναμεναμε. Τώρα από την άλλη για μνήμη του FPGA παρατηρούμε ότι δεν πήρε κανέναν πόρο και όλη η ενεργεια που χρησιμοποιεί είναι static οπότε λογικά ή είναι αρκετά ελλιπής η περιγραφή του κυκλώματος μας έτσι ώστε ο synthesizer να μην μπορεί να υλοποιήσει κάτι(αρκετά πιθανό αφού όπως ανέφερα η περιγραφή του κυκλώματος την μνήμης RAMB36 είναι ελλιπής ή απλά αξιοποίησε κάποιο Block RAM όπου ταιριαζει είναι αρκετά κοντά στην περιγραφή τους.

## Κώδικες σε VHDL

Παρακάτω παρατίθενται οι κώδικες της εργασίας:

---

### RAM

---

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.NUMERIC_STD.all;

ENTITY RAM IS
  GENERIC(A: INTEGER:=5;
          n: INTEGER:=8);
  PORT(
    WR:   IN STD_LOGIC; --WR=1 Write Enable allws Read Enable
    ADDR: IN STD_LOGIC_VECTOR(A-1 DOWNTO 0); --RAM dieuthinseis
    DIN:  IN STD_LOGIC_VECTOR(n-1 DOWNTO 0); --Write Dedomena
    DOUT: OUT STD_LOGIC_VECTOR(n-1 DOWNTO 0)); --Read Dedomena
END ENTITY RAM;

ARCHITECTURE RAMbehavior OF RAM IS
  SUBTYPE word IS STD_LOGIC_VECTOR(n-1 DOWNTO 0); --Kathorise to megethos ths lekshs
  TYPE          MEMORY is ARRAY(0 to 2**A-1) OF word; --Megethos ths Mnhmhs
  SIGNAL        RAM_32_BYTES: MEMORY; --RAM_32_BYTES ws shma toy typoy MEMORY

  BEGIN
    PROCESS(WR,DIN,ADDR)
      VARIABLE RAM_ADDR_IN: NATURAL RANGE 0 to 2**A-1; --Dieuthinseis se akereous
      VARIABLE INITIALIZE: BOOLEAN:=TRUE; --metablhth gia arxikopoihsh ths mnhmhs
    BEGIN
      RAM_ADDR_IN:=TO_INTEGER(UNSIGNED(ADDR)); --Metatropi dieuthinsewn se akeraious
      IF(INITIALIZE=TRUE) THEN
        RAM_32_BYTES<=(0=>"00000001",
                        1=>"00000010", --Arxikopouhsh ths mnhmhs bazontas ti
                        2=>"00000011", --Stis protes 4 thesis ths mnhmhs >0
                        3=>"00000100", --Kai se oles tis alles 0
                        OTHERS=>"00000000");
        DOUT<="XXXXXXXX"; --Mh kathorismenes times sthn eksodo ths RAM
        INITIALIZE:=FALSE; --H arxikopoihsh etsi ekteleite
                                --mono mia fora sthn arxh
      ELSIF (WR='1') THEN --Diadikasias Eggrafhs Dedomenwn
        RAM_32_BYTES(RAM_ADDR_IN)<=DIN; --Synexeia anagnosis
      END IF;
    END PROCESS
  END ARCHITECTURE
```

```

        DOUT<=RAM_32_BYTES(RAM_ADDR_IN); --Sunexeia Anagnosis
    END PROCESS;

END ARCHITECTURE RAMbehavior;

```

---

## RAMB36

---

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;
USE IEEE.NUMERIC_STD.all;

ENTITY RAMB36 IS
    GENERIC(A: INTEGER:=5;
            n: INTEGER:=8);
    PORT(
        WRADDR: IN STD_LOGIC_VECTOR(A-1 DOWNTO 0);
        RDADDR: IN STD_LOGIC_VECTOR(A-1 DOWNTO 0);
        DI: IN STD_LOGIC_VECTOR(n-1 DOWNTO 0);
        DO: OUT STD_LOGIC_VECTOR(n-1 DOWNTO 0);
        RDCLK: IN STD_LOGIC;
        WRCLK: IN STD_LOGIC;
        WREN: IN STD_LOGIC;
        RDEN: IN STD_LOGIC);
END ENTITY RAMB36;

ARCHITECTURE RAMbehavior OF RAMB36 IS
    SUBTYPE word IS STD_LOGIC_VECTOR(n-1 DOWNTO 0); --Kathorise to megethos ths lekshs
    TYPE MEMORY is ARRAY(0 to 2**A-1) OF word; --Megethos ths Mnhmhs
    SIGNAL RAM_32_BYTES: MEMORY; --RAM_32_BYTES ws shma toy typoy MEMORY

BEGIN
    PROCESS(DI, RDADDR, RDCLK, WRCLK, WREN, RDEN, WRADDR)
        VARIABLE RAM_ADDR_WRIN: NATURAL RANGE 0 to 2**A-1; --Dieuthinseis se akereon
        VARIABLE RAM_ADDR_RDIN: NATURAL RANGE 0 to 2**A-1;
        VARIABLE INITIALIZE: BOOLEAN:=TRUE; --metablhth gia arxikopoihsh ths mnhmhs
    BEGIN
        RAM_ADDR_WRIN:=TO_INTEGER(UNSIGNED(WRADDR)); --Metatropi dieuthinsewn se akeraious
        RAM_ADDR_RDIN:=TO_INTEGER(UNSIGNED(RDADDR)); --Metatropi dieuthinsewn se akeraious
        IF(INITIALIZE=TRUE) THEN
            RAM_32_BYTES<=(0=>"00000001",
                            1=>"00000010", --Arxikopouhsh ths mnhmhs bazontas tis
                            2=>"00000011", --Stis protes 4 thesisi ths mnhmhs >0
                            3=>"00000100", --Kai se oles tis alles 0
                            OTHERS=>"00000000");
        END IF;
    END PROCESS;

```

```

DO<="XXXXXXXX"; --Mh kathorismenes times sthn eksodo ths RAM
INITIALIZE:=FALSE; --H arxikopoihsh etsi ekteleite
--mono mia fora sthn arxh
ELSIF (WREN='1' AND WRCLK'EVENT' AND WRCLK='1') THEN --Diadikasias Eggrafhs De
    RAM_32_BYTES(RAM_ADDR_WRIN)<=DI;
ELSIF(RDEN = '1' AND RDCLK'EVENT' AND RDCLK='1') THEN
    DO<=RAM_32_BYTES(RAM_ADDR_RDIN); --Sunexxia Anagnosis
END IF;
END PROCESS;

END ARCHITECTURE RAMbehavior;

```

## Βιβλιογραφία

- [1] Xilinx, Inc., *Artix-7 FPGA Data Sheet: DC and AC Switching Characteristics*, 2024. DS181.
- [2] Xilinx, Inc., *7 Series FPGAs Memory Resources User Guide*, 2019. UG473.